

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A computer-implemented method for identifying user interface (UI) objects in a markup-language stream, the method comprising the steps of:  
receiving, from a server and at a computer system, a web-based application for display in a web browser, the web-based application comprising one or more web pages;

receiving a predefined grammar for ~~a particular~~ the web-based application;  
automatically generating a parser computer program based on the predefined grammar using an automated parser generator tool;

scanning ~~(i) the markup-language stream or (ii) a corresponding~~ document object model (DOM) of the web-based application with the parser computer program to generate tokens;

parsing the tokens with the parser computer program to identify at least ~~one UI objects in a portion of the particular graphical element in the web-based application;~~ and  
outputting the portion of the particular application- one or more UI objects that correspond to the at least one graphical element in the web-based application.

2. (Original) The method of claim 1, wherein said markup-language stream drives a markup-language-based browser application, and wherein the scanning step includes scanning the DOM generated by a browser that displays that application.

3. (Original) The method of claim 1, wherein the scanning step includes identifying elements of the DOM by traversal thereof.

4. (Canceled)

5. (Previously Presented) The method of claim 3, wherein the scanning step includes generating one or more tokens for each scanned DOM element.

6.-7. (Canceled) .

8. (Currently Amended) The method of claim 1, wherein the at least one UI objects comprises one of a user input ~~fields~~, ~~text fields~~, ~~metatags~~, field, a text field, a metatag, unprintable markup-language, or an in-line ~~images~~, image.

9. (Currently Amended) The method of claim 1, wherein the scanning and parsing steps are adapted to identify UI objects that correspond to elements displayed in the ~~markup-language~~ web-based application.

10. (Currently Amended) The method of claim 1, further comprising grouping the tokens into syntactic structures that identify items displayed by the ~~particular web-based~~ web-based application.

11. (Previously Presented) The method of claim 10, wherein said step of grouping comprises identifying similarly formatted markup-language elements based on their markup-language attributes such as classname, font size, style, tag color, and size.

12. (Previously Presented) The method of claim 1, wherein said at least one UI objects comprises a name, content, a shape, or a location.

13. (Previously Presented) The method of claim 1, wherein automatically generating said the parser computer program comprises executing YACC ("Yet Another Compiler-Compiler").

14.-15. (Canceled).

16. (Previously Presented) The method of claim 1, wherein the parser computer program is a LALR(1) parser.

17. (Previously Presented) The method of claim 1, wherein the parser computer program is a LR(1) parser.

18. (Previously Presented) The method of claims 1, wherein the markup language is any of HTML, XHTML and XUL.

19. (Currently Amended) A digital data processing system comprising:  
a client digital data processor configured to:  
receive, from a server and at a computer system, a web-based application for display in a web browser, the web-based application comprising one or more web pages;

~~receiving-receive~~ a predefined grammar for a particular application;

receive a predefined grammar for ~~a particular~~ the web-based application;

automatically generate a parser computer program based on the predefined grammar using an automated parser generator tool;

~~scan the (i) the markup language stream or (ii) a corresponding~~  
document object model (DOM) of the web-based application with the parser computer program to generate tokens;

parse the tokens with the parser computer program to identify at least one ~~UI objects in a portion of the particular~~ graphical element in the web-based application; and

~~output the portion of the particular application;~~ one or more UI objects that correspond to the at least one graphical element in the web-based application.

20. (Canceled)

21. (Currently Amended) The digital data processing system of claim 20, wherein said one or more UI objects each comprise name, content, shape, location, and properties.

22. (Canceled).

23. (Currently Amended) The digital data processing system of claim 19, wherein said tokens are interpreted according to the predefined grammar to identify and distinguish among UI objects of ~~a markup language~~ the web-based application's display.

24. (Currently Amended) The digital data processing system of claim 19, wherein the at least one UI object comprises a user input ~~fields, field,~~ a text ~~fields, field,~~ a ~~metatags, metatag,~~ unprintable markup-language, or an in-line ~~images, image.~~

25. (Previously Presented) The digital data processing system of claim 19, wherein the markup language is any of HTML, XHTML and XUL.

26. (Currently Amended) The method of claim 1, further comprising providing context-based help based at least in part on the at least one graphical element in the web-based ~~portion of the particular~~ application.

27. (Currently Amended) The digital data processing system of claim 19, wherein the client digital data processor is further configured to provide context-based help based at least in part on the at least one graphical element in the web-based ~~portion of the particular~~ application.